In the claims

Please replace the originally filed claims with the claims listed below. The claims listed below are marked to show the differences from the originally filed claims.

1. (Cancelled)

2. (Currently Amended) A hardware computing machine, which will be referred to as an Effector machine, comprising:

(a) a collection of hardware computing elements, which will be referred to as Effectors, that are each communicatively coupled to at least one other Effector; and

(b) a machine architecture that

adjusts how the Effectors behave and

adjusts how information is transmitted from one Effector to another Effector~~The machine of claim 1~~ wherein a subset of said Effectors ~~are~~ is configured to receive information from a Static program.

3. Cancelled.

4. Cancelled.

5. (Currently Amended) A system comprising a computer readable medium storing thereon one or more instructions that constitute an input interpreter for designing at least

a hardware computing machine, which will be referred to as an Effector machine, including at least

(a) a collection of hardware computing elements, which will be referred to as Effectors, that are each communicatively coupled to at least one other Effector, and

(b) a machine architecture that

adjusts how the Effectors behave and

adjusts how information is transmitted from one Effector to another Effector~~the machine of claim 1,~~; and

the input interpreter outputs a software Effector machine, which is a design for the hardware Effector machine.

6. (Previously presented) The system of claim 5 wherein

the software Effector machine is a first software Effector machine,

said input interpreter is implemented with a second Effector machine that includes at least a second collection of software computing elements.

7. Cancelled.

8. (Currently Amended) A system comprising:

the Effector machine of claim ~~1~~37, and

an output interpreter in addition to the Effector machine, wherein the interpreter is for translating firing activity of a subset of said Effectors into a desired output form.

9. (Previously presented) The system of claim 8 wherein

the collection of hardware computing elements is a first collection of hardware computing elements,

the Effector machine is a first Effector machine and said output interpreter is implemented with a second Effector machine that is constructed from a second collection of hardware computing elements, and

the first collection of hardware elements and the second collection of hardware elements are different subsets of a third collection of hardware computing elements.

10. (Currently Amended) The machine of claim ~~1~~ 37 wherein said machine is a dynamic machine in that one or more parameters of the Effectors are functions of time.

11. (Currently Amended) The machine of claim 10 wherein the machine is for running a Meta program that changes, over time, one or more properties associated with one or more of said Effectors, the ~~meta~~Meta program being a sequence of sets, each set being a list of values of parameters of Effectors, and each list of values having the parameters,

the machine including at least a portion for receiving the ~~meta~~Meta program and for converting the ~~meta~~Meta program into input for that machine.

12. (Currently Amended) The machine of claim ~~1~~ 37 wherein said machine architecture comprises hardware having a predetermined error tolerance ~~by~~ limiting a range of values to which one or more parameters of the hardware are allowed to be set.

13. (Currently Amended) The machine of claim 12 wherein the hardware includes transistors configured to operate at or below a gate voltage at which the mobile charge in the transistor begins to limit the flow of current, which is called the threshold voltage.

14. (Currently Amended) A method comprising designing a machine~~the machine of claim 1~~, at least ~~partially,~~ by evolving a graph representing the machine to produce a design of the ~~effector~~ machine,

the machine being a hardware computing machine, which will be referred to as an

Effector machine, including at least

(a) a collection of hardware computing elements, which will be referred to as

Effectors, that are each communicatively coupled to at least one other

Effector; and

(b) a machine architecture that

adjusts how the Effectors behave and

adjusts how information is transmitted from one Effector to another

Effector.

15. (Previously Presented) The method of claim 14 wherein the evolving of the graph includes at least performing a crossover of two representations of Effector machines via

interchanging representations of modules of the two representations of Effector machines.

16. (Currently Amended) The method of claim 14 wherein the evolving of the graph is for changing the following properties:

a number of software modules per software machine,

a number of software ~~effector~~Effectors per software module,

one or more refractory periods associated with one or more software ~~effector~~Effectors,

one or more thresholds associated with one or more software ~~effector~~Effectors,

a number of software connections,

one or more amplitudes associated with two or more software ~~effector~~Effectors,

one or more pulse widths associated with two or more software ~~effector~~Effectors, and

one or more conduction times associated with two or more software ~~effector~~Effectors.

17. (Cancelled)

18. (Currently Amended) The method of claim ~~17~~ 23 wherein a subset of said Effectors, called Input Effectors, are for receiving information from an external environment.

19. Cancelled.

20. Cancelled.

21. (Currently Amended) The method of claim ~~17~~23 further comprising designing the Effector machine via an input interpreter.

22. Cancelled.

23. (Currently Amended) A method, comprising: providing a hardware computing machine, which will be called an Effector machine, by at least

(a) providing a collection of hardware computing elements, which will be referred to as Effectors,

(b) communicatively coupling each Effector of the collection to at least one other Effector;

(c) providing a machine architecture that, while the machine is running,

adjusts how Effectors behave, and

adjusts how information is transmitted from one Effector to another Effector;

~~T~~the method ~~of claim 17,~~ further comprising designing said machine architecture by at least evolving a graph associated with the machine architecture.

24. (Currently Amended) The method of claim ~~17~~26, further comprising configuring a subset of said Effectors, called Output Effectors, for translating at least one firing activity of the Output Effectors into a desired output form.

25. (Previously Presented) The method of claim 24 wherein

the collection of hardware computing elements is a first collection of hardware computing elements,

the Effector machine is a first Effector machine and the translating is performed via an output interpreter that is implemented with a second Effector machine that is constructed from a second collection of hardware computing elements, and

the first collection of hardware elements and the second collection of hardware elements are different subsets of a third collection of hardware computing elements.

26. (Currently Amended) The method of claim ~~17~~ 23 wherein the machine architecture and ~~effector~~Effectors are part of a dynamic machine in which parameters of the Effectors are functions of time.

27. (Currently Amended) The method of claim 26, wherein the dynamic machine is for running a Meta program that changes, over time, one or more properties associated with one or more of the Effectors, the Meta program being a sequence of sets, each set being a list of values of parameters of Effectors, and each list of values having the parameters in a set order, the machine including at least a portion for receiving the ~~meta~~Meta program and converting the ~~meta~~ Meta program into input ~~that~~ for the machine.

28. (Currently Amended) The method of claim ~~17~~23, further comprising designing said machine architecture to limit values to which one or more parameters are allowed to be set, based on an error tolerance.

29. (Currently Amended) The method of claim 28 wherein said designing includes at least configuring transistors to operate at subthreshold based on the error tolerance limiting a range of values that at least one parameter of the Effectors is allowed to be set, operating at subthreshold refers to operating below a gate voltage at which the mobile charge in the transistor begins to limit the flow of current, which is called the threshold voltage.

30. Cancelled.

31. (Previously Presented) The method of claim 23 further comprising crossing over representations of modules of Effector machines between two representations of Effector machines associated with the graph.

32. (Currently Amended) The method of claim 23 wherein evolving the graph includes at least changing one or more of the following properties associated with at least a portion of a representation of the machine: a number of software modules per software machine, a number of software ~~effector~~Effectors per software module, a refractory period associated with at least one of the software Effectors, a threshold associated with a software Effector, a number of software connections between the software Effectors, an amplitude associated with one or more of the software Effectors, a pulse width associated with one or more of the software Effectors, and a conduction time between at least two of the software Effectors.

33. Cancelled.

34. (Currently Amended) The method of claim ~~17~~ 23 further comprising designing a least one circuit that is associated with the machine by at least evolving a graph associated with the circuit.

35. (Currently Amended) The machine of claim ~~1~~ 37 wherein a subset of said Effectors are configured to receive information from an external environment.

36. (Currently Amended) The machine of claim ~~1~~ 37 wherein

the collection of hardware computing elements is a first collection of hardware computing elements,

the Effector machine is a first Effector machine, ~~and~~

a subset of said Effectors are configured to receive information from a second Effector machine that is constructed from a second collection of hardware computing elements, and

the first collection of hardware elements and the second collection of hardware elements are different subsets of a third collection of hardware computing elements.

37. (Currently Amended) A hardware computing machine, which will be referred to as an Effector machine, comprising:

(a) a collection of hardware computing elements, which will be referred to as Effectors, that are each communicatively coupled to at least one other Effector; and

(b) a machine architecture that

adjusts how the Effectors behave and

adjusts how information is transmitted from one Effector to another Effector~~The machine of claim 1~~; wherein a subset of said Effectors are configured to receive information from a Meta program, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

38. (Currently Amended) The machine of claim ~~1~~ 37 wherein the ~~effector~~Effector machine is a first ~~effector~~Effector machine, a subset of said Effectors are for receiving information from an external environment;

the subset of said Effectors are for being configured to receive information from a second Effector machine;

the subset of said Effectors are for being configured to receive information from a Static program;

the subset of said Effectors are for being configured to receive information from a Meta program, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors; and

the subset of said Effectors are for being configured to receive information from any combination of the external environment, the second Effector machine, Static program, and the Meta program.

39. (Previously Presented) The method of claim 14, wherein said evolving of the graph includes at least changing a number of representations of modules in a representation of the machine.

40. (Currently Amended) The method of claim 14, wherein said evolving of the graph includes at least changing a number of software ~~effector~~Effectors per software module.

41. (Previously Presented) The method of claim 14, wherein said evolving of the graph includes at least changing one or more refractory periods associated with one or more software Effectors.

42. (Previously Presented) The method of claim 14, wherein said evolving of the graph includes at least changing one or more thresholds associated with one or more software Effectors associated with the graph.

43. (Previously Presented) The method of claim 14, wherein said evolving of the graph includes at least changing a number of software connections between two or more software Effectors.

44. (Previously Presented) The method of claim 14 wherein said evolving of the graph includes at least changing one or more representations of amplitudes associated with one or more representations of Effectors associated with the graph.

45. (Previously Presented) The method of claim 14 wherein said evolving of the graph includes at least changing one or more representations of pulse widths associated with representations of the Effectors.

46. (Previously Presented) The method of claim 14 wherein said evolving of the graph includes at least changing one or more representations of conduction times associated with representations of the Effectors.

47. (Currently Amended) The machine of claim ~~1~~2, further comprising an input interpreter for ~~at least partially~~ designing at least athe Static program for the Effector machine.

48. (Currently Amended) The machine of claim ~~1~~37, further comprising an input interpreter for ~~at least partially~~ designing at least a Meta program for the Effector machine, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

49. (Currently Amended) The machine of claim 10 wherein the machine is for running a Meta program that changes, over time, one or more properties of said machine, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors, the machine including at least a portion for receiving the ~~meta~~ Meta program and converting the ~~meta~~ Meta program into input for that machine.

50. (Currently Amended) The method of claim ~~17~~ 23 wherein the Effector machine is a first Effector machine and a subset of said Effectors, called
Input Effectors, are for receiving information from a second Effector machine.

51. (Currently Amended) The method of claim ~~17~~23 wherein a subset of said Effectors, called Input Effectors, are for receiving information from a Static program.

52. (Currently Amended) The method of claim ~~17~~ 23 wherein a subset of said Effectors, called Input Effectors, are for receiving information from a Meta program, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

53. (Currently Amended) The method of claim ~~17~~ 23 wherein the Effector machine is a first Effector machine, a subset of said Effectors, called Input Effectors, are for receiving information from an external environment;
the Input Effectors are for receiving information from a second Effector machine;
the Input Effectors are for receiving information from a Static program; and
the Input Effectors are for receiving information from a Meta program, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

54. (Currently Amended) The method of claim 26 wherein the dynamic machine is for running a Meta program that changes, over time, a threshold associated with one or more

Effectors, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

55. (Currently Amended) The method of claim 26 wherein the dynamic machine is for running a Meta program that changes, over time, a refractory period associated with one or more Effectors, the ~~meta~~ Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

56. (Previously Presented) The method of claim 26 wherein the dynamic machine is for running a Meta program that changes, over time, a pulse amplitude associated with two or more Effectors, the Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

57. (Previously Presented) The method of claim 26 wherein the dynamic machine is for running a Meta program that changes, over time, a pulse width associated with two or more Effectors, the Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

58. (Previously Presented) The method of claim 26 wherein the dynamic machine is for running a Meta program that changes, over time, a transmission time associated with two or more Effectors, the Meta program being a sequence of sets, each set being a list of values of parameters of Effectors.

59. (Previously Presented) The system of claim 8, the system being configured such that the interpreter interprets whether an Effector fires as binary information, and interprets the binary information into the desired output form, which includes at least a sequence of symbols.

60. (Previously Presented) A machine readable medium storing instructions for configuring one or more digital computers to run a software machine, the software machine comprising:

(a) a collection of software computing elements, which will be referred to as software Effectors, that are each communicatively coupled to at least one other software Effector;

(b) a machine architecture that

determines how the software Effectors behave, and

determines how information is transmitted from one software Effector to another software Effector;

(c) the software machine receives input instructions in a language native to the software machine and outputs instructions in a language native to the digital computer.

61. (Previously Presented) A method, comprising forming a hardware computing machine by at least:

(a) providing a collection of hardware computing elements, which will be referred to as Effectors,

(b) providing a machine architecture that

determines how Effectors behave and

determines how information is transmitted from one Effector to another Effector;

(c) configuring each Effector of the collection to be communicatively coupled to at least one other Effector, and

(d) configuring a portion of the hardware computing machine for receiving input that sets values for one or more parameters of individual Effectors from the collections of Effectors, the one or more parameters including a time at which information is transmitted from the individual Effectors to another of the individual Effectors.

62. (Currently Amended) A hardware computing machine, which will be referred to as an Effector machine, comprising:

(a) a collection of hardware computing elements, which will be referred to as Effectors, each Effector of the collection being communicatively coupled to at least one other Effector;

(b) a machine architecture that, while the hardware computing machine is running

determines how the Effectors behave and

determines how information is transmitted from one Effector to another Effector; and

(c) the hardware computing machine including a portion for receiving input that sets values for one or more parameters of individual Effectors from the collections of Effectors, the one or more parameters including a time at which information is transmitted from ~~of~~ the individual Effectors to another of the individual Effectors.